

Řízení polohovacích úloh pomocí programovatelného automatu

Control of Positioning Tasks using a Programmable Controller

Vojtěch Vavrečka

Bakalářská práce

Vedoucí práce: prof. Ing. Jiří Koziorek, Ph.D.

Ostrava, 2021

Abstrakt

Cílem této bakalářské práce je seznámení se s možnostmi polohování pomocí programovatelných automatů. Teoretická část obsahuje analýzu možností polohování servopohonu, použití dodatečných snímačů a řízení servopohonu. Experimentální část práce zachycuje laboratorní instalaci pro ukázkou polohování servopohonu, včetně zhodnocení rizik a návrhu opatření pro zajištění funkční bezpečnosti. Dále je zde popsáno testování možností polohování na laboratorní instalaci a postup při tvorbě řídicí aplikace v prostředí BOSCH Rexroth Indra Works Engineering pro úlohu představující ovládání tří simulovaných výtahů. Pro tuto výslednou úlohu je popsán konkrétní postup vytvoření komunikace mezi PLC a vizualizací. Vytvoření samotného uživatelského rozhraní bylo provedeno za pomoci vizualizace na HMI panelu, která slouží pro ovládání vytvořené aplikace.

Klíčová slova: servopohon, možnosti polohování, PLC, řídicí aplikace, uživatelské rozhraní

Abstract

The aim of this bachelor thesis is to get acquainted with the possibilities of positioning using a programmable controller. The theoretical part contains an analysis of the possibilities of positioning the servomotor, the use of additional sensors and control of the servomotor. The experimental part of the work describes a laboratory installation to demonstrate the positioning of the servomotor, including risk assessment and suggestions of measures to ensure functional safety. It also describes the testing of positioning options for laboratory installation and the procedure for creating a control application in the BOSCH Rexroth Indra Works Engineering environment for a task representing the control of three simulated elevators. For this final task, a specific procedure for creating communication between the PLC and the visualization is described. The creation of the user interface itself was accomplished using visualization on the HMI panel, which is used to control the created application.

Keywords: servomotor, possibilities of positioning, PLC, control application, user interface

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce **prof. Ing. Jiřímu Koziorkovi, Ph.D.** za vedení a průběžné konzultace během vypracovávání této bakalářské práce. Dále bych také chtěl poděkovat **Ing. Martinu Mikolajkovi** za pomoc při návrhu laboratorní instalace a panu **Lumírovi Malickému** za pomoc při samotné konstrukci laboratorní instalace.

Obsah

Seznam použitých zkratk a symbolů	6
Seznam obrázků	7
Seznam tabulek.....	8
Úvod	9
1 Servopohon	10
1.1 Enkodéry	10
1.1.1 Rotační optický enkodér	11
1.2 Řízení pomocí frekvenčních měničů	12
1.3 Způsoby zjišťování polohy	13
1.4 Typy polohování	14
2 PLCopen	15
2.1 Činnosti organizace	15
2.2 Programovací jazyky	15
2.3 Technické zaměření	16
2.4 Motion control	17
2.4.1 Stavový diagram	18
3 Návrh a realizace laboratorní instalace pro prezentaci možností polohování... 19	19
3.1 PLC Bosch Rexroth Indra Control L45.....	20
3.2 Servopohon MS2N	21
3.3 Frekvenční měnič IndraDrive HCS01	22
4 Zhodnocení rizik a návrh opatření pro zajištění funkční bezpečnosti	23
4.1 Posouzení rizika podle ČSN ISO 13849-1.....	23
4.1.1 Performance Level.....	24
4.2 Návrh opatření	25
5 Návrh a realizace řídicí aplikace pro laboratorní instalaci	26
5.1 HW konfigurace PLC.....	27
5.2 Komunikace pomocí Sercos III	28
5.3 Nastavení parametrů reálných os	29
5.3.1 Testování polohování	29
5.3.2 Nastavení limitů.....	30
5.4 Části řídicí aplikace.....	31
5.4.1 Definice proměnných	32
5.4.2 Program pro vyhodnocování	34
5.4.3 Program pro řízení servopohonu/výtahu	36

6	Návrh a realizace uživatelského rozhraní	37
6.1	HW konfigurace HMI panelu	37
6.2	Konfigurace komunikace	37
6.3	Použité tagy ve vizualizaci	38
6.4	Použité objekty a funkce	39
6.5	Výsledná realizace uživatelského rozhraní	40
	Závěr.....	41
	Literatura.....	42
	Seznam příloh	44

Seznam použitých zkratk a symbolů

CoDeSys	Controlled Development System
EtherCAT	Ethernet for Control Automation Technology
FBD	Function Block Diagram
IL	Instruction List
LD	Ladder Diagram
PID regulátor	Proportionální, integrační, derivační regulátor
PLC	Programmable Logic Controller
Profibus	Process Field Bus
SERCOS	Sercos Automation Bus
SFC	Sequential Function Chart
ST	Structured Text
XML	Extensible Markup Language

Seznam obrázků

Obrázek 1 Disk inkrementálního enkodéru [4]	10
Obrázek 2 Disk absolutního enkodéru [2]	11
Obrázek 3 Rotační optický enkodér [3]	11
Obrázek 4 Ukázka indukčního snímače polohy [8]	13
Obrázek 5 Jazyk LD – Ladder Diagram [14]	16
Obrázek 6 Jazyk FBD – Function Block Diagram [15]	16
Obrázek 7 Použití XML formátu [21]	17
Obrázek 8 Stavový diagram – PLCopen Motion control [25]	18
Obrázek 9 Navržena laboratorní instalace	19
Obrázek 10 Servopohon MS2N03 [27]	21
Obrázek 11 Frekvenční měniče řady HCS01 [28]	22
Obrázek 12 Graf rizik – Performance Level	24
Obrázek 13 Vývojové prostředí Indra Works	26
Obrázek 14 HW konfigurace PLC	27
Obrázek 15 Nastavení parametrů osy	29
Obrázek 16 Nastavení limitů osy	30
Obrázek 17 Rozdělení táskového systému	31
Obrázek 18 Blokové schéma programu pro vyhodnocení	35
Obrázek 19 Stavový diagram programu řízení	36
Obrázek 20 Konfigurace OPC-UA serveru	37
Obrázek 21 Symbol configuration	38
Obrázek 22 Ovládací panel výtahů	40

Seznam tabulek

Tabulka 1 Technické parametry – PLC L45 [26].....	20
Tabulka 2 Technické parametry – Servopohon MS2N03 [27].....	21
Tabulka 3 Technické parametry – Frekvenčního měniče HCS01.1E-W0009 [28]	22
Tabulka 4 Zvolení normy podle použité technologie [29].....	23
Tabulka 5 Seznam použitých globálních proměnných v projektu	32
Tabulka 6 Seznam použitých lokálních proměnných v projektu	33
Tabulka 7 Seznam použitých tagu ve vizualizaci	38

Úvod

Bakalářská práce je zaměřena na řízení polohovacích úloh pomocí programovatelných automatů. Vzhledem k stále většímu zájmu o vytváření automatických strojů je zapotřebí se věnovat správnému polohování a řízení servopohonu. Teoretický úvod práce obsahuje analýzu možností polohování servopohonů, a to za použití technologií zakomponovaných přímo v těle servopohonu, a jejich následné využití v praxi. Dále je zde popsáno řešení určování polohy pomocí snímačů, samotné řízení servopohonu pomocí frekvenčních měničů a typy polohování. Nalezneme zde popis používaných programovacích jazyků, které jsou definovány ve standardu organizace PLCopen. Podrobněji se věnuji jedné z činností technického zaměření organizace PLCopen, a to programování „Motion aplikací“.

Pro samotnou prezentaci možností polohování byla navržena a vyrobena laboratorní instalace s využitím komponent od firmy Bosch. V práci je popsán návrh laboratorní instalace, který byl vytvořen tak, aby sloužil nejen pro potřeby bakalářské práce, ale také pro následnou výuku studentů. Nechybí zde popisy technických parametrů použitých komponent. Při návrhu laboratorní instalace je potřeba vzít v úvahu možná rizika, která mohou vzniknout při práci na tomto zařízení. Potřebám zajištění funkční bezpečnosti a samotnému stanovení výsledného rizika je v bakalářské práci věnována samostatná kapitola.

Finálním krokem bakalářské práce po dokončení realizace laboratorní instalace je samotné uvedení úlohy do provozu. Na vytvořené úloze byly vyzkoušeny a otestovány možnosti polohování za účelem správného výběru polohování pro použití ve výsledné řídicí aplikaci. Řídicí aplikace pro ovládání tří simulovaných výtahů je popsána v kapitole č.5. Řídicí aplikace je vytvořena v prostředí BOSCH Rexroth Indra Works Engineering. Součástí této bakalářské práce je vytvoření uživatelského rozhraní pro ovládání této úlohy, včetně vytvoření komunikace mezi vizualizací a PLC. Vytvoření samotného uživatelského rozhraní bylo provedeno pomocí vizualizace na HMI panelu, která slouží pro ovládání vytvořené aplikace.

1 Servopohon

Servopohon je lineární nebo rotační pohon, který na rozdíl od klasického motoru umožňuje přesné nastavení polohy. Hlavní výhodou je jednoduché a velmi přesné nastavení potřebné polohy natočení osy nebo posunu. S využitím těchto pohonů se můžeme setkat například v automatizované výrobě, robotice nebo u CNC strojů.

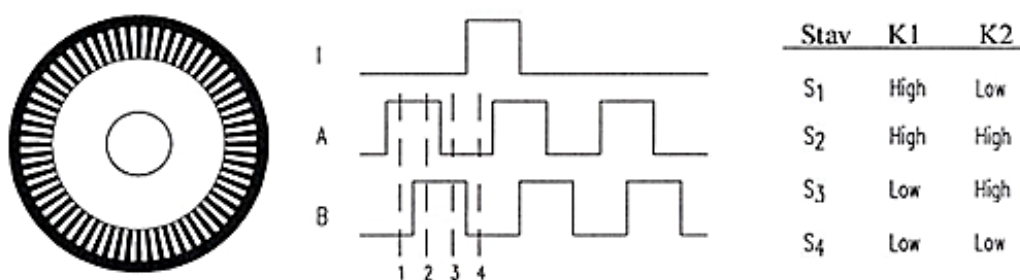
Jedná se o stejnosměrný nebo střídavý motor, který ve spojení s uzavřenou zpětnou vazbou tvoří servopohon. Ten pak může být doplněn ještě o brzdu, která slouží k udržení nastavené polohy osy. Zpětná vazba servopohonu slouží k okamžitému vyhodnocení pohybu nebo polohy osy a k okamžité regulaci rychlosti pohybu nebo polohy, na rozdíl od krokových motorů, které zpětnou vazbu nemají. Jejich ovládání se provádí v takzvaných krocích, zpravidla natočení o určitý úhel/krok.

Primitivní servopohony využívají pro sledování polohy potenciometr a ovládání je prováděno pouze dvěma stavy, tedy rotací plnou rychlostí nebo zastavením. Tyto jednoduché servopohony se v průmyslu tolik nevyužívají, ale u RC modelů jsou velmi rozšířené díky své nízké ceně. Dokonalejší servopohony k řízení otáček motoru využívají rotační optický enkodér. Tento druh servopohonu využívá k řízení proměnné otáčky pro přesnější nastavení polohy. [1]

1.1 Enkodéry

Enkodér je zařízení, které se stará o převod mechanického pohybu – buď pozice hřídele nebo úhlu natočení na elektrický signál. Využití těchto enkodérů není jenom v servopohonech, ale téměř ve všech aplikacích robotiky, kde se využívají servopohony. Výstupem enkodérů je elektrický signál, jehož hodnota vyjadřuje momentální polohu, úhlovou rychlost otáčení nebo zrychlení. Tato hodnota se využívá k řízení aplikace.

Rotační enkodéry můžeme rozdělit na inkrementální, které generují sled pulsů, a absolutní enkodéry, které vytváří sled unikátního kódů. Inkrementální enkodér generuje pulzy při rotačním pohybu, což odpovídá určitému natočení hřídele nebo posuvu. Rychlost se pak vyhodnocuje podle počtu pulsů za určitý čas.



Obrázek 1 Disk inkrementálního enkodéru [4]

Absolutní enkodéry mají předem definované číselné hodnoty, které reprezentují konkrétní úhel natočení. Enkodér tedy generuje číselnou hodnotu, pomocí které vyhodnotíme přesně danou pozici hřídele. V aplikacích se můžeme často setkat s použitím rotačního optického enkodéru. [2][3]

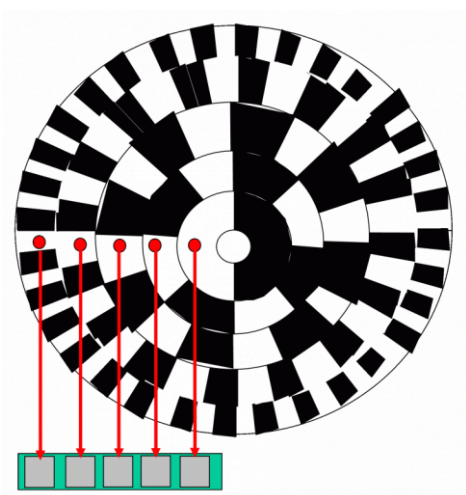
1.1.1 Rotační optický enkodér

Zjednodušeně řečeno můžeme považovat rotační optický enkodér za elektromechanický převodník, který se stará o převod mechanického pohybu na sekvenci elektrických digitálních impulsů. Rotační optický enkodér se skládá ze tří hlavních částí. Disk s opticky průhlednými, nebo odraznými a neprůhlednými, anebo neodraznými ploškami. Zdrojem světla bývá nejčastěji LED dioda, která vysílá světlo ve viditelném nebo infračerveném spektru. Fototranzistor nebo fotodioda pak plní funkci optického přijímače.

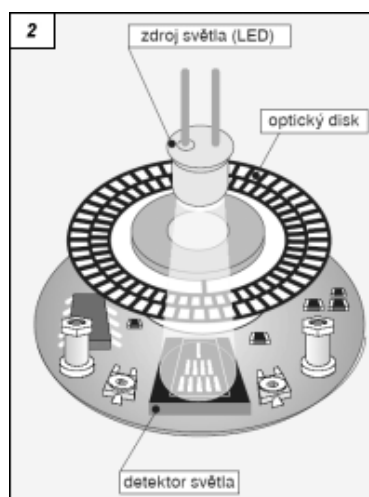
Disk je připevněn k hřídeli enkodéru a současně k hřídeli pohonu, což zaručí synchronní pohyb disku a hřídele pohonu. Světelný zdroj umístěný před diskem vyzařuje světlo, které prochází diskem v opticky průhledných místech a v opticky neprůhledných je světlo pohlceno. Otáčení disku generuje světelné impulsy, které zachycuje optický přijímač umístěný na druhé straně disku, který převádí tyto světelné impulsy na impulsy elektrické. Pro eliminaci případného rušení musí být signál zesílen a elektricky zpracován.

U enkodéru reflexních je světelný zdroj a optický přijímač na stejné straně disku. Na disku jsou odrazné a matné plošky, které při otáčení disku před přijímačem způsobují změnu napětí podle množství detekovaného odraženého světla. Výstupem převodu je obdélníkový signál. Rovnoměrné rozdělení odrazných a matných plošek zaručuje posun vždy o stejný počet stupňů, těmto enkodérům říkáme inkrementální.

Dalším typem enkodéru je absolutní enkodér. Ten na rozdíl od inkrementálního, který detekuje pouze změnu o daný krok, detekuje přesně danou polohu v rámci 360 stupňů. Tohoto jevu je docíleno zakódováním každého určitého úhlu jedinečnou kombinací pomocí Grayova kódu. Čím více bitů má enkodér, tím je rozlišitelnost vyšší (jemnější rozdělení kotouče). [2][3]



Obrázek 2 Disk absolutního enkodéru [2]



Obrázek 3 Rotační optický enkodér [3]

1.2 Řízení pomocí frekvenčních měničů

Frekvenční měnič slouží k regulaci otáček elektromotorů, čehož je docíleno změnou frekvence a napětí, nebo pouze změnou frekvence. Toto řízení se používá tam, kde je potřeba plynule nebo skokově měnit rychlost pohonu. Taktéž se tento způsob řízení často využívá pro řízení rozběh a doběh motorů, abychom docílili výrazného snížení proudových a mechanických rázů. Frekvenční měniče jsou určeny primárně pro řízení třífázových elektromotorů. Střídavé napájecí napětí může být jednofázové, nebo třífázové napětí. Frekvenční měnič pomocí vnitřních obvodů napájecí napětí usměrní a na výstupu měniče je napětí převedeno na třífázové napětí o požadované frekvenci pro řízení elektromotorů. Ovládání měničů je možno mnoha způsoby: manuální nebo automatické ovládání pomocí externích ovládacích prvků nebo dálkové řízení pomocí PLC nebo automatická regulace pomocí zpětné vazby. V automatizaci je nejvíce využíváno řízení pomocí PLC nebo jiných řídicích systémů nebo automatická regulace, kde se o řízení stará PID regulátor. [5]

Frekvenční měniče pro programovatelné logické automaty využívají komunikaci mezi PLC a měničem a následně získanými instrukcemi z PLC je řízen servopohon. Aktuální stavy pohonu, jako jsou aktuální poloha či rychlost, jsou předány do PLC, které vykonává veškeré řízení pohonu. Hlavní výhodou je řízení více pohonu pomocí jednoho PLC. [6]

1.3 Způsoby zjišťování polohy

Dalším způsobem pro zjišťování polohy servopohonu jsou snímače. Tyto snímače jsou připojeny přímo do PLC nebo frekvenčního měniče a slouží nejčastěji jako koncové snímače polohy neboli jinak řečeno referenční snímače. Pro tyto účely se volí snímače nespojitě, které vysílají digitální signál 0 nebo 1. Mezi nejpoužívanější snímače patří odporové, kapacitní a indukčnostní. Všechny tyto uvedené typy mohou fungovat také ve spojitěm režimu. Tento režim slouží pro zjišťování přesné polohy posuny osy, většinou v lineárním směru. Tyto snímače vrací spojitou elektrickou veličinu neboli analogový signál.[7][9]

Odporový nespojitý snímač funguje na principu skokové změny odporu, přepínání kontaktů, například v provedení pružinového spínače. Používán je nejčastěji jako snímač dojezdu nebo referenční senzor. U spojitěho odporového snímače je využit potenciometr nebo trimr, kde běžec je připojen na osu pohybu, a tím dochází ke změně odporu. [7]

Kapacitní snímače fungují na principu změny kapacity v okolí snímače pro vyhodnocení polohy. U nespojitěho snímače se nastaví citlivost sepnutí a u spojitěho se vyhodnocuje změna kapacity. Výhodou těchto snímačů je bezkontaktní měření, a to jakéhokoliv materiálu. [7]

Indukčnostní snímače fungují na principu změny indukčnosti cívky, která je způsobena pohybem jádra v cívce. Toto jádro bývá kovový předmět spojený s osou servopohonu. Tento typ snímačů může pracovat také v bezkontaktním režimu, ale reaguje pouze na kovové předměty. Také může tento snímač být použit jako spojitý i nespojitý. [7]



Obrázek 4 Ukázka Indukčního snímače polohy [8]

1.4 Typy polohování

Pro každou polohovací úlohu je třeba zvolit typ polohování pro programování dané osy. Pro soustavu os lze využít pohledu na úlohu jako na kinematiku, což pomáhá při programování složitějších úloh. Mezi tyto kinematiky patří například delta nebo x-y-z manipulátor. Kinematiky jsou soustavy složené z jednotlivých typů polohování. Základní typy polohování jsou následující:

- Absolutní polohování – Pro tento druh polohování je důležité nastavení referenčního bodu, jenž se většinou nastavuje pomocí referenčního snímače, který pro danou osu definuje nulu. Toto polohování se používá pro ovládání lineárních os.
- Polohová tabulka – Tento typ polohování se používá pro úlohy, kde je přesně daná poloha, například naložení výrobků, pracovní stanoviště a vyložení výrobků. Pro tyto polohy je vytvořená tabulka hodnot a přiřazeno nastavení pomocí bitů.
- Rotační osa – Tento druh polohování osy se používá pro úlohy s otočným mechanismem, robotické rameno, otočný karusel. Ovládání osy se provádí nastavením úhlů natočení servopohonu, většinou 0 až 360°.
- Translační osa – Translační osa se používá pro zjednodušení polohování u os se závitovou tyčí. Jedná se o převod mezi posunem na ose a otočením pohonu, a to nastavením konstanty „k“.

2 PLCopen

PLCopen je organizace podporující efektivitu průmyslové automatizace. Organizace vznikla v důsledku stále vzrůstající automatizace, kdy vznikaly nové nekompatibilní software a nové funkce, které způsobovaly zdržení vývoje a zvyšovaly náklad na vývoj aplikace. Toto vedlo ke vzniku organizace PLCopen a definování normy IEC 61131. [10]

2.1 Činnosti organizace

Organizace se zabývá problémy a potřebami uživatelů a je závislá na podpoře uživatelů pomocí víceúrovňových členských příspěvků, od firemních členských příspěvků až po vzdělávací instituce. Organizace PLCopen se zabývá mezinárodním standardem IEC 61131-3, který patří pod normu IEC 61131. Díky použití tohoto standardu jsou uživatelé schopni jednodušeji vyvíjet své aplikace pro průmyslovou automatizaci, což vede ke snížení nákladů při vývoji a také k úspoře času.

Pro ještě větší pohodlí při vývoji vznikají standardizované knihovny pro práci v různých odvětvích průmyslové automatizace a rozhraní pro transparentní komunikaci. Organizace PLCopen společně s koncovými uživateli také definuje otevřené standardy pro efektivnější vývoj. Použití těchto standardů usnadňuje upgrade, případné rozšíření dané aplikace a také snadnější komunikaci s ostatními software. Další velkou výhodou je velké množství členů a uživatelů používající tuto otevřenou platformu a díky tomu se můžeme spolehnout na obrovskou podporu v případě problémů při vývoji aplikace. [10][11]

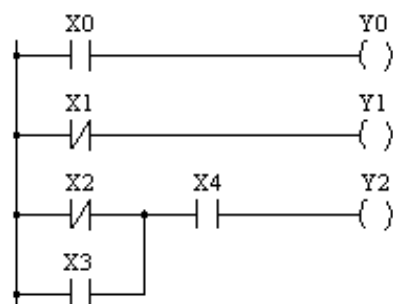
2.2 Programovací jazyky

Další důležitou částí normy je definování používaných programovacích jazyků pro automatizaci. Standard IEC 61131-3 obsahuje pět definovaných programovacích jazyků, které můžeme spíše znát pod označením CoDeSys, univerzální vývojové prostředí pro vývoj řídicích aplikací pro PLC, bez ohledu na použitý hardware. Těchto pět jazyků má přesně definovanou strukturu, kterou není možno upravovat. Jednotlivé jazyky mají mezi sebou souvislosti, a dokonce některé vývojové prostředí je umí mezi sebou převádět. Těchto pět jazyků, které se nachází ve standardu jsou: [12][13]

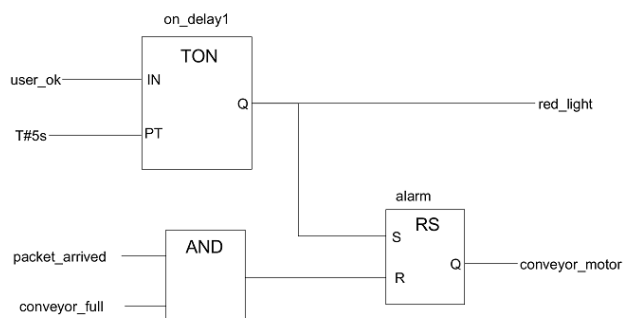
- IL - Instruction List
- ST - Structured Text
- LD - Ladder Diagram
- SFC - Sequential Function Chart
- FBD - Function Block Diagram

Jazyk IL je základní textový programovací jazyk, který je velmi podobný jazyku assembler (ASM) používaný u mikrokontrolérů. Dalším textovým programovacím jazykem je ST (tzv. Strukturovaný text), což je už vyšší programovací jazyk, který je velmi vhodný pro klasické programátory, protože má velkou podobnost v jazyce Pascal nebo C. Vzhledem k snadné a rychlé realizaci i složitých úloh je tento jazyk velmi výkonný a hojně využívaný zejména při psaní funkčních bloků. Další programovací jazyky řadíme mezi tzv. „grafické programovací jazyky“. Mezi tyto grafické programovací jazyky patří v první řadě jazyk pod názvem Ladder Diagram (LD), jinak řečeno reléové schéma. Je založený na soustavě propojených relé, pomocí kterých se vytváří jednotlivé logické funkce. Je to velmi rozšířený programovací jazyk, který je oblíbený zejména díky své přehlednosti a rychlému zpracování logických signálů. Dalším z graficky orientovaných jazyků je jazyk SFC (Sequential Function Chart), používaný pro jednoduché definování chování programu prostřednictvím vývojových diagramů, které znázorňují rozhodovací strukturu, která vychází z binárních Petriho sítí. Posledním jazykem, který řadíme také

mezi grafické jazyky je FBD (tzv. schéma funkčních bloků). Jedná se o obdobu jazyku LD, kde místo sériového/paralelního zapojení relé jsou postupně propojené bloky, které představují jednotlivá hradla, nebo logické operace. [12][13]



Obrázek 5 Jazyk LD – Ladder Diagram [14]



Obrázek 6 Jazyk FBD – Function Block Diagram [15]

2.3 Technické zaměření

Technické zaměření je tvořeno šesti bloky, které mezi sebou úzce souvisí. Názvy bloků jsou následující: logika, kontrola pohybu, bezpečnost, komunikace, výměna XML a certifikace. [16]

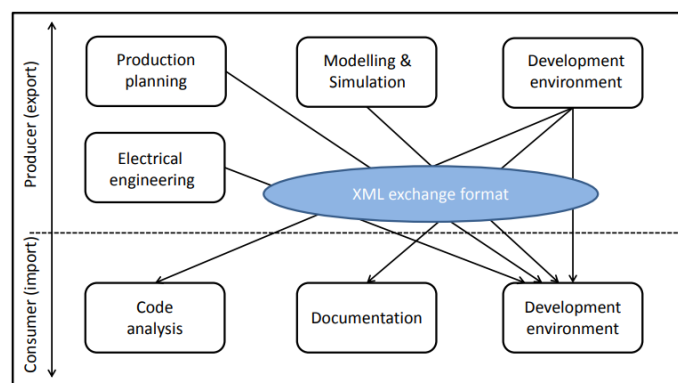
Blok logika vychází z již mnohokrát zmiňovaného mezinárodního standardu IEC 61131 a jeho části 3, která je zaměřena na programovací jazyky. Zabývá se oblastí syntaktických a sématických pravidel, sady programovacích prvků a implementace daných programovacích jazyků do průmyslových aplikací. [17]

Blok kontrola pohybu obsahuje sadu specifikací a funkcí pro práci s polohovacími úlohami. Detailněji popsáno v kapitole 2.4 Motion Control.

Sada specifikací PLCopen bezpečnost obsahuje funkční bloky a pokyny pro uživatele, které napomáhají při vývoji logických a pohybových aplikací zároveň s důležitými bezpečnostními prvky. Díky certifikaci této sady je časově a finančně méně náročné pro příslušné organizace certifikace jejich vyvíjených aplikací. Pro bezpečnost v oboru řízení pohonů a pohybu, kde se stále častěji používají příkazy přes síťové systémy, byly vytvořené obecné návrhy řešení řízení pohybu v těchto sítích jako jsou například ProfiSafe, Safety over Ethercat, CIP Safety over Sercos a OpenSafety. [18]

Vzhledem k stále rozšiřujícímu se trendu s Průmyslem 4.0, internetem věcí a Smart Factories je komunikace klíčovým prvkem. Je stále větší zájem o poskytování více a více funkcí a tím přibývá i potřeba připojovat více a více zařízení do tzv. inteligentních systému. Organizace PLCopen společně s nadací OPC Foundation využila pro toto řešení standard OPC UA, který je komunikačním protokolem v komunikaci mezi stroji v průmyslové automatizaci. Díky tomuto standardu, který je nad rámec normy IEC61131-3, se mohou úspěšně rozvíjet tzv. inteligentní systémy. [19]

Stále častější potřeba uživatelů vyměňovat si své programy, knihovny a projekty mezi sebou, a to napříč vývojovými prostředími, vedla ke vzniku bloku výměna XML. XML je značkovací jazyk hojně používaný při tvorbě webových stránek. Výsledkem řešení organizace PLCopen je převedení projektu bez ztráty informací, i když je neúplný nebo obsahuje syntaktické chyby. Hlavním cílem je, aby tento nástroj nesloužil pouze pro export a import mezi vývojovými prostředími, ale aby taky sloužil jako rozhraní pro jiné softwarové nástroje například pro tvorbu dokumentací, modelování nebo simulace (viz. Obrázek 6). [20]



Obrázek 7 Použití XML formátu [21]

Posledním zaměřením je již zmíněna certifikace. Jedním z aspektů práce PLCopen je certifikace programovacích systémů v souladu s normou IEC61131. Certifikace se provádí hlavně pro bloky logika, kontrola pohybu a bezpečnost. [22]

2.4 Motion control

Motion control neboli taky v překladu ovládání či kontrola pohybu je jedním z technického zaměření organizace PLCopen. Hlavním důvodem vzniku zaměření motion control je používání různých hardware pro řízení pohybu. Před vznikem organizace PLCopen bylo potřeba ke každému použitému hardware vytvořit jedinečný software, i když funkce různých hardware byly totožné. [23]

Pohybový standard od organizace PLCopen proto přišel s řešením, jak mít standardní aplikační knihovny, které lze využít pro různé hardware bez nutnosti opakovaného vytváření dané knihovny. Hlavní výhodou je snížení času a také nákladů na vývoj a také poskytuje lepší údržbu daného systému. Zároveň vývojoví inženýři nejsou tolik nuceni s každým dalším hardwarem se učit něco nového, nebo dokonce být posíláni na školení. Samozřejmě standardizované knihovny neobsahují veškeré funkce, ale pouze často používané funkce. Díky tomuto řešení se stává programování méně závislé na hardwarem a zvyšuje se opětovné použití aplikačního softwaru. [24]

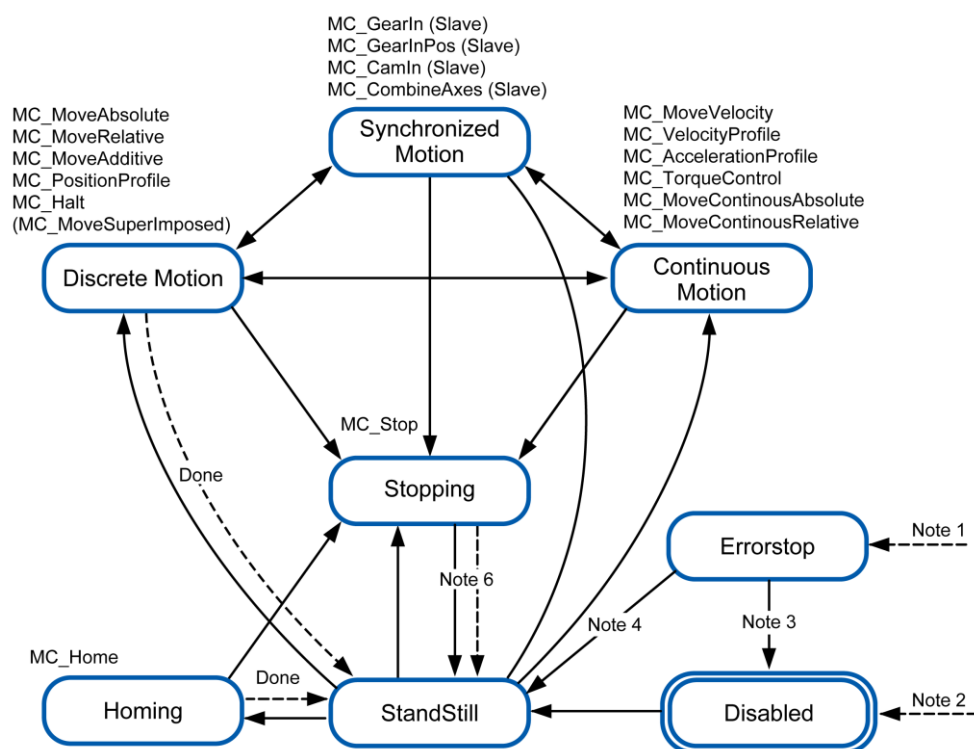
Nyní v sada specifikací řízení pohybu podle organizace PLCopen obsahuje:

- Funkční bloky pro řízení pohybu
- Pokyny pro uživatele
- Koordinovaný pohyb
- Postupy pro referencování (Homing)
- Rozšíření

2.4.1 Stavový diagram

Při tvorbě řídicích aplikací se začíná definicí stavového automatu, který slouží pro grafické znázornění vývoje systému. Dalším krokem je definice sady základních funkčních bloků pro řízení osy. Důležité je také dodržování předpisů pro případnou certifikaci aplikace. [23]

Díky pevné struktuře stavového diagramu, který definuje jasný přechod mezi jednotlivými stavy a je nutné s tímto počítat při psaní programu. Pro všechny stavy jsou také připravené dané funkce. Vzhled stavového diagramu a jednotlivých funkcí je znázorněn na obrázku 8.



Obrázek 8 Stavový diagram – PLCopen Motion control [25]

Doplnění popisu stavového diagramu: [25]

- Note 1 – Zjištění chyby (přechod z jakéhokoliv stavu)
- Note 2 – Nebyla zjištěna chyba (přechod z jakéhokoliv stavu)
- Note 3 – MC_Reset and MC_Power.Status = FALSE
- Note 4 – MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = TRUE
- Note 5 – MC_Power.Enable = TRUE and MC_Power.Status = TRUE
- Note 6 – MC_Stop.Done = TRUE and MC_Stop.Execute = FALSE

3 Návrh a realizace laboratorní instalace pro prezentaci možností polohování

Pro ukázkou možnosti řízení polohovacích úloh pomocí servopohonu byla navržena laboratorní instalace, která byla umístěna a zakomponována v místnosti Smartfactory – CPIT TL3. Laboratorní instalace byla navržena tak, aby následně sloužila jako výuková úloha pro studenty vysoké školy oboru řídicí a informační systémy. Navržená instalace obsahuje tři na sobě nezávislé osy, které jsou řízeny pomocí jednoho průmyslového automatu od firmy Bosch. Tyto osy představují simulaci tří výtahů. Každý z těchto tří výtahů je poháněn jedním servomotorem. Výtah je simulován vyznačením kabiny výtahu na pohybujícím se řemenu. Laboratorní instalace nemusí představovat jen úlohu výtahů, ale je připravena na prezentaci možností ovládání servopohonu. Je možné si vyzkoušet polohování absolutně či relativně, nebo nastavování úhlu natočení osy. Dále je možno ovládat osy jednotlivě, nebo následně synchronizace více servopohonu. Díky vytvoření úlohy výtahů pomocí řemenů, je úloha velmi vhodná pro výuku programování pohybových úloh, aniž by došlo ke zničení či poškození úlohy v případě chyby v programování.

Pro laboratorní instalaci byl navržen uzavíratelný box z hliníkových profilů, do kterého byla úloha nainstalována pro zajištění bezpečnosti při práci. Čelní panel navrženého boxu byl navrhnut z průhledného polykarbonátu. Zadní strana byla vyřešena pomocí uzamykatelných dveří, kde byla umístěna veškerá elektrická zařízení. V laboratorní instalaci byl použit PLC automat od firmy Bosch. PLC od této firmy bylo vybráno pro velké použití v pohybových úlohách v průmyslu. Taktéž servopohony a frekvenční měniče byly použity od této firmy. Konkrétně PLC Bosch Rexroth Indra Control L45, servopohony s označením MS2N03 a měniče s označením IndraDrive HCS01.



Obrázek 9 Navržená laboratorní instalace

3.1 PLC Bosch Rexroth Indra Control L45

Výhody použitého PLC:

- Škálovatelná hardwarová platforma
- Palubní komunikační rozhraní
- Volitelné rozšíření prostřednictvím funkčních a technologických modulů
- Modulární Input a Output jednotky

Technické parametry:

Tabulka 1 Technické parametry – PLC L45 [26]

Procesor	
CPU	AMD-Geode LX800, 500MHz
Hodiny reálného času	integrated
Paměť	
Aplikace	256 MB
Remanentní paměť	256 kB
Vyrovňovací paměť	8 MB
Flash paměť	1 GB
Rozhraní	
Ethernet	1 x Ethernet TCP/IP (Standard)
Ready	1.0 x Ready contact (Standard)
Serial	2.0 x Ethernet TCP/IP (Optional)
Input/Output	
Digital inputs	8x DC oddělené vstupy
Digital outputs	8x DC oddělené výstupy
Použité kanály	Max. 256
Funkční moduly	Max. 4
Elektrické údaje	
Napájecí napětí U	24 V DC
$U_{min} \dots U_{max}$	19 V DC ... 30 V DC
Logické napájení U_L	7,5 V
	Max. 2 A
Logické segmentové napájení U_{LS}	Tolerance -25 % ... +30 %
	Max. 3 A
Segmentové napájecí napětí U_s	Tolerance -15 % ... +20 %
	Max. 8 A
Hlavní napájecí napětí U_M	24 V DC
	Tolerance -15 % ... +20 %
	Max. 8 A

3.2 Servopohon MS2N

Pro laboratorní úlohu byly vybrány synchronní servopohony od firmy Bosch řady MS2N, konkrétně typ MS2N03-B0BYN. Jedná se o menší kompaktní servopohon bez brzdy s rychlostí 9000 otáček za minutu s vlastním chlazením. Použitý rotační enkodér u tohoto typu je pokročilý víceotáčkový absolutní enkodér z rozlišitelnosti 4096 otáček, jedná se o digitální 20-ti bitový enkodér. Instalace servopohonu je velmi jednoduchá díky propojení servopohonu s řídicí jednotkou pohonu frekvenčním měničem pomocí jediného kabelu. Jedná se o inteligentní servopohon, který poskytuje data o jednotlivých odečtech, které jsou uloženy v datové paměti motoru. Řídicí jednotka pohonu, měnič IndraDrive, velmi rychle zpracovává tyto data a tím mnohonásobně zvyšuje přesnost točivého momentu a snižuje se tak chybovost na zlomek hodnot. Díky poskytnutí těchto dat lze tímto způsobem vytvořit aplikace v rámci Průmyslu 4.0 efektivně a bez nutnosti použití dalších komponent. Model pohonu MS2N v konstrukčním nástroji IndraSize umožňuje jednoduchou, rychlou a bezpečnou konfiguraci pohonu, která odpovídá přesně skutečnému provozu. Integrované enkodéry zajišťují maximální bezpečnost pro aplikace SafeMotion. [27]

Tabulka 2 Technické parametry – Servopohon MS2N03 [27]

Základní parametry	
Napájecí napětí U	115 V nebo 230 V
Maximální proud I_{MAX}	7,25 A
Točivý moment	3,75 Nm
Maximální rychlost	9000 ot/min
Brzda	NE
Enkodér	
Typ enkodéru	víceotáčkový – absolutní
Rozlišitelnost	4096 otáček
Provedení	Digitální – 20 bitů



Obrázek 10 Servopohon MS2N03 [27]

3.3 Frekvenční měnič IndraDrive HCS01

V laboratorní úloze byly použity pro ovládání servopohonu frekvenční měniče řady IndraDrive HCS01, konkrétně typ HCS01.1E-W0009. Mezi hlavní funkční parametry tohoto typu měniče patří:

- Analogový vstup (14 bitů, ± 10 V)
- 8 digitálních vstupů – jeden kombinovaný, který lze konfigurovat jako digitální vstup nebo jako digitální výstup
- Komunikace na bázi Ethernetu, podpora protokolů: Sercos III, PROFINET IO, EtherNet/IP, EtherCAT
- Integrovaná bezpečnostní technologie: Safe Motion, L3 (Safe Torque Off), L4 (Safe Torque Off, Safe Brake Control) [28]

Tabulka 3 Technické parametry – Frekvenčního měniče HCS01.1E-W0009 [28]

Elektrické údaje	
Síťové napájecí napětí	3 AC 110 ... 230 V
Maximální proud	9 A
Rozhraní	
Komunikace	Multi-Ethernet, včetně Sercos III
Input/Output	
Digital inputs	8x
Kombinovaný Input/Output	1x
Analog input	1x (14 bitů, ± 10 V)



Obrázek 11 Frekvenční měniče řady HCS01 [28]

4 Zhodnocení rizik a návrh opatření pro zajištění funkční bezpečnosti

Při návrhu a realizaci laboratorní úlohy je potřeba vzít v úvahu rizika, která by mohla vzniknout při práci na laboratorní úloze. Analýza rizik slouží pro posouzení rizika a umožňuje rozhodnutí, zda je nebo není toto riziko potřeba snížit. V případě vysokého rizika je potřeba použít bezpečnostní řešení. Možným řešením pro zvýšení bezpečnosti, které nabízí přímo firmy vyrábějící průmyslové automaty. Ať už to je samotné Safety PLC, nebo doplňujících modulu safety a jejich bezpečnostní prvky jako je stop tlačítko, nebo případné světelné závory pro zabránění vniku nepovolané osoby do nebezpečného prostoru.

4.1 Posouzení rizika podle ČSN ISO 13849-1

Jedná se o normu, která je nástupcem normy ČSN EN 954-1. Norma definuje požadavky a návrh na provedení bezpečnostní části řídicího systému. Pro určení bezpečnostní úrovně je v této normě využíváno vyhodnocení na základě metody PL – Performance level. Jedná se o diskrétní úrovně, které jsou definovány jako pravděpodobnost poruchy za hodinu.

Další velmi používanou normu pro vyhodnocení funkční bezpečnosti je norma ČSN EN 62 061. Mezi těmito dvěma normami existuje určitý spojovací prvek, a to je jejich vyhodnocovací parametry, z tohoto důvodu je zakázáno tyto normy kombinovat. Správné zvolení normy závisí na použité technologii v systému (Tabulka 4). [29]

Tabulka 4 Zvolení normy podle použité technologie [29]

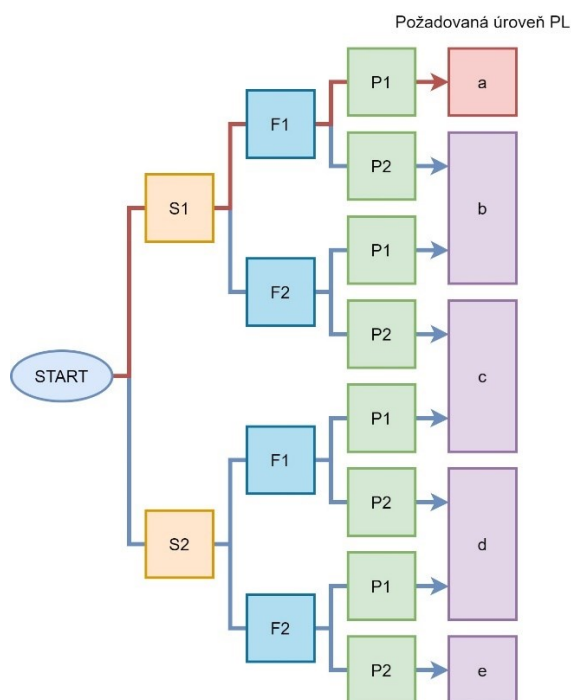
Technologie	ČSN ISO 13849-1	ČSN EN 62 061
Hydraulika	Použitelná	Není možné použít
Pneumatika	Použitelná	Není možné použít
Mechanika	Použitelná	Není možné použít
Elektrika	Použitelná	Použitelná
Elektronika	Použitelná	Použitelná
Programovatelná elektronika	Použitelná	Použitelná

4.1.1 Performance Level

Stanovení Performance Level, dále už jen PL, vychází z normy ČSN ISO 13849-1. PL jsou definovány do pěti úrovní (úroveň PLa až PLe), kde úroveň PLa zastupuje nízké riziko a naopak vysoké riziko zastupuje úroveň PLe. Vyhodnocení se provádí na základě grafu rizik v několika krocích, graf viz. obrázek 12. [29]

Jednotlivé kroky pro určení PL úrovně:

- Závažnost zranění – S:
 - S1 = lehké, bez následků zranění
 - S2 = vážné, nevratné zranění včetně smrti
- Četnost nebo doba vystavení nebezpečí – F:
 - F1 = zřídka nebo krátká doba
 - F2 = častá až trvalá nebo dlouhá doba
- Možnosti vyloučení nebezpečí – P:
 - P1 = možné vyhnout se nebezpečí za určitých podmínek
 - P2 = téměř nemožné se vyhnout nebezpečí



Obrázek 12 Graf rizik – Performance Level

Po vyhodnocení jednotlivých kroků, kde závažnost zranění byla stanovena S1, doba vystavení se nebezpečí F1 a možnost vyloučení nebezpečí byla určena P1. Z těchto hodnot byla stanovena úroveň bezpečnosti na hodnotu PLa. Toto stanovení a postup v grafu je zaznačen červenou barvou na obrázku 12 výše.

4.2 Návrh opatření

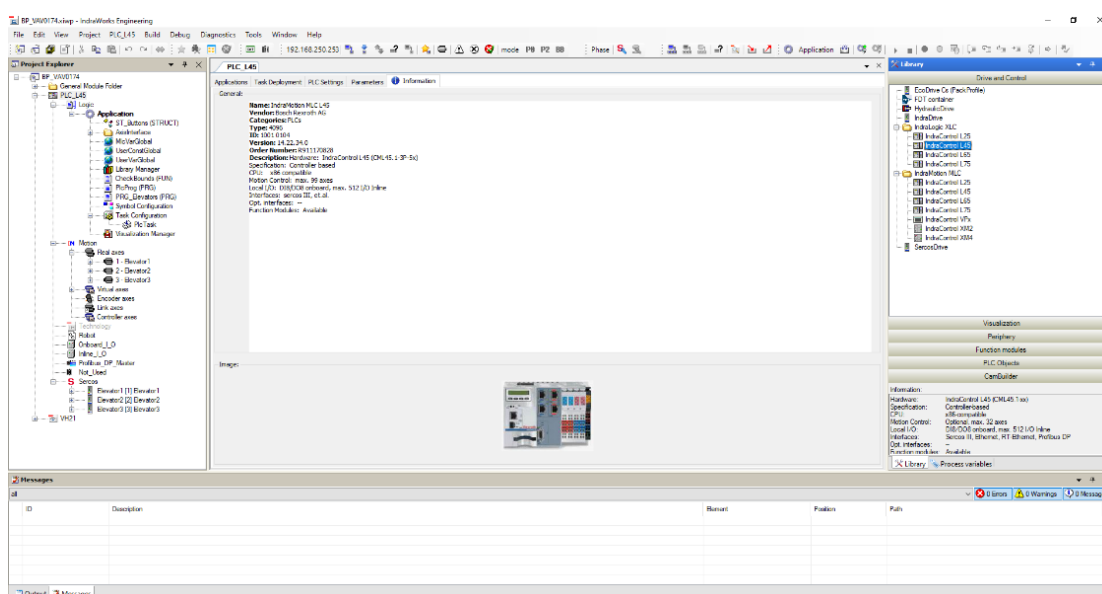
Hlavním opatřením pro zajištění funkční bezpečnosti je celkové konstrukční provedení laboratorní instalace, která je umístěna v uzavíratelném boxu zkonstruovaného z hliníkových profilů. Čelní strana boxu je vyrobena z průhledného polykarbonátu. Průhledný polykarbonát kromě toho, že umožňuje pozorovat nainstalovanou úlohu v chodu, plní také ochrannou funkci. Veškerá elektrická zařízení, která jsou potřebná pro chod laboratorní instalace, jsou umístěna uvnitř tohoto boxu. Případný přístup pro potřebné úpravy nebo změny umožňují uzamykatelné dveře, které zároveň zabraňují vniknutí nepovolaným osobám k zařízením umístěným uvnitř boxu.

Druhým navrženým opatřením je použití vypínače napětí, který je umístěný na vnější straně uzavíratelného boxu. V případě jakékoliv poruchy na laboratorní instalaci nám tento vypínač slouží jako nouzové vypnutí, kdy dojde k odpojení elektrické energie pro celou laboratorní instalaci.

5 Návrh a realizace řídicí aplikace pro laboratorní instalaci

Pro vývoj řídicí aplikace byl použit software BOSCH Rexroth Indra Works Engineering. Programování PLC a řízení pohybových úloh je založené na CoDeSys V3. Softwarový nástroj slouží pro vývoj programu, nastavení pohonů, komunikaci a kompletní diagnostiku. Nástroj zahrnuje také komplexní softwarové knihovny podle IEC 61131-3 a PLCopen a podporuje komunikaci Ethernet v reálném čase, včetně SERCOS III a Profinet. Dále tento software nástroj umožňuje vytvářet vizualizace pro panely HMI pro tvorbu uživatelského rozhraní. Popřípadě je možné využít pro tvorbu vizualizace externí program WinStudio, který je specializován přímo pro tvorbu vizualizací.

Pro tvorbu aplikace byl zvolen textový jazyk ST, který je vhodný pro ovládání a řízení os z důvodu již importované knihovny pro tvorbu motion aplikací s funkcemi právě v tomto jazyce. Dalším důvodem použití právě tohoto jazyka byla dřívější zkušenost a pohodlí s programováním právě v tomto jazyce.

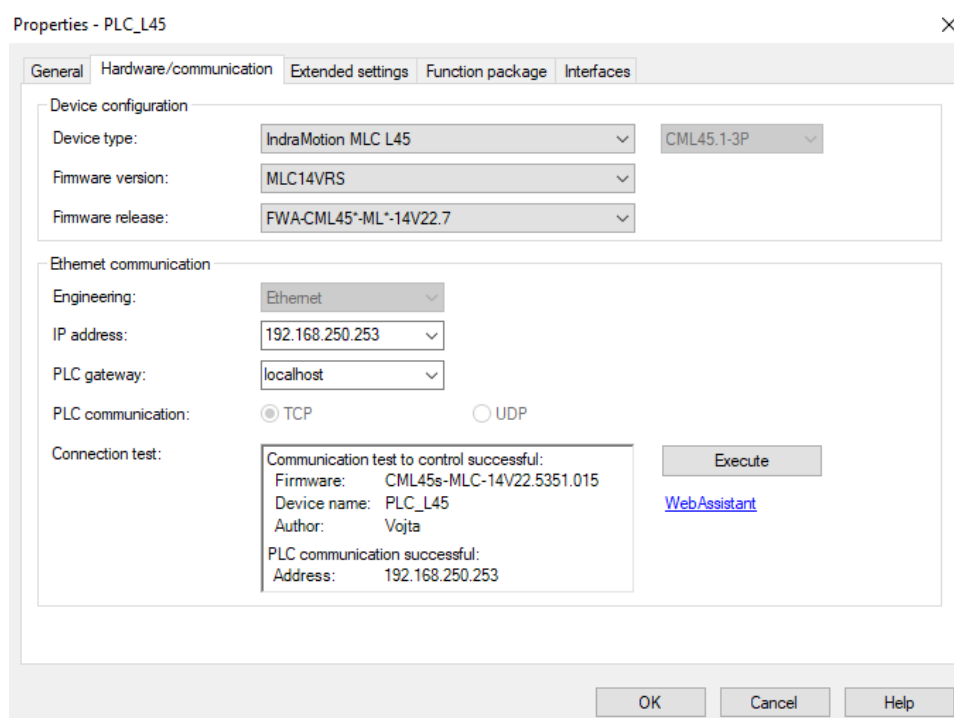


Obrázek 13 Vývojové prostředí Indra Works

5.1 HW konfigurace PLC

Po založení projektu je zapotřebí provést hardwarovou konfiguraci všech použitých zařízení. Pro tento projekt bude stačit přidat pouze PLC a pak HMI panel pro tvorbu vizualizace. Bylo tedy přidáno PLC IndraMotion MLC L45 a proveden upgrade verze firmwaru v PLC na verzi 14V22. Je zapotřebí sjednotit verzi firmwaru PLC s verzemi firmware projektů, aby bylo možné pokračovat v následné práci. Na displeji PLC bylo nastavena IP adresa 192.168.250.253 s maskou podsítě 255.255.255.0. Tato adresa byla společně s nastavením komunikace na Ethernet provedena při konfiguraci PLC v projektu. Poté je možné ověřit komunikaci s PLC na síti. V případě problému s připojením se k PLC je možné pro zjištění problému využít WebAssistant. Jeho využití je převážně u novější řady PLC automatu, kde již nenalezneme displej pro konfiguraci, a je tedy nutné provést nastavení adresy právě pomocí tohoto asistenta.

Z PLC řady MLC byly vybrány funkční balíčky – logická část PLC, motion kontrol, technologické funkce a robot kontrol. V případě použití jiné řady PLC je zapotřebí pro vývoj této aplikace vybrat balíček logická část a motion kontrol. Další fází konfigurace by bylo přidání všech dodatečných připojených karet, jako jsou vstupní a výstupní karty. Použité PLC má vstupní a výstupní karty, ale není zapotřebí je přidávat jako další periferie, ale jsou již rovnou zahrnuty v konfiguraci přidáním PLC z knihovny.



Obrázek 14 HW konfigurace PLC

Další částí konfigurace je přidání HMI panelu pro vizualizaci. Jelikož k úloze nebude připojen panel HMI fyzicky, ale vizualizace bude spuštěna na připojeném notebooku, tak nemusel být vložený přesný model panelu. Podrobnější informace ohledně přidání a konfigurace komunikace HMI panelu je popsáno v kapitole č.6.

5.2 Komunikace pomocí Sercos III

Sběrnice Sercos III je průmyslová komunikační sběrnice, která bude sloužit pro komunikaci mezi PLC a frekvenčními měniči. Sercos sběrnice je vlastní sběrnici firmy Bosch, která poskytuje real-time řízení. Pro správnou komunikaci je třeba vytvořit ringovou topologii zapojení neboli zapojení zařízení do kruhu pomocí ethernet kabelu. V případě rozpojení je ringová topologie rozdělena na dvě části, kdy vzniknou dvě topologie Bus.

Pro přidání zařízení do této sítě je potřeba zařízením přidělit koncové adresy vytvořené sítě 172.31.254. Toto nastavení bylo provedeno na displeji frekvenčního měniče, nebo je také možnost změnit adresu zařízení po nalezení dostupných zařízení pomocí skenování sítě Sercos v prostředí IndraWorks. Frekvenčním měničům byly přiděleny adresy 1,2,3 a následně byly přidány do sítě. Sít Sercos obsahuje množství komunikačních fází sloužících k nastavování, konfiguraci, kontrole a běhu připojených modulů. Jednotlivé fáze lze přepínat pro celou Sercos síť, nebo pro jednotlivé zařízení v síti zvlášť.

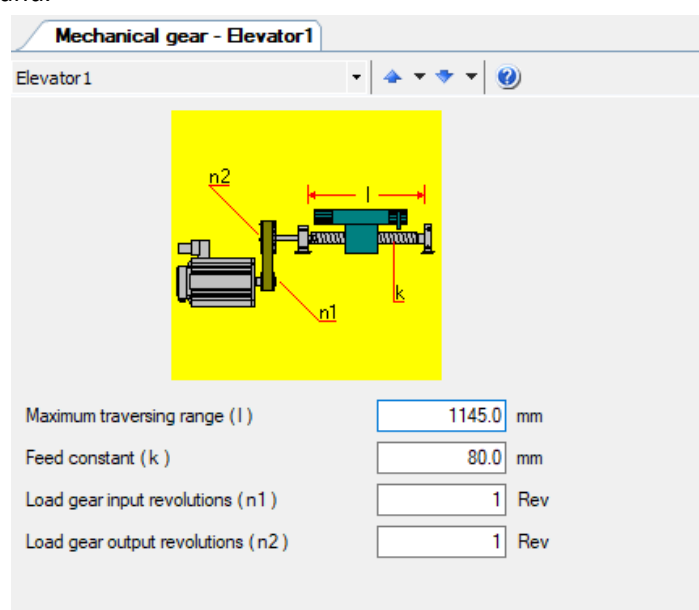
Komunikační fáze sběrnice Sercos III:

NRT	Klasická Ethernet komunikace dostupná.
P0	Navázání komunikace mezi účastníky Sercosu. Test ringové topologie pomocí synchronizačního signálu. Sken sběrnice.
P1	Identifikace účastníků Sercosu
P2	Příprava časové a datové struktury protokolů, parametrizační mód. Možnost nastavení parametrů jednotlivých Sercos účastníků.
P3	Kontrola platnosti parametrů. Poslední synchronizační test se slave.
P4 (BB)	Provozní stav. Realtime přenos je aktivní.

5.3 Nastavení parametrů reálných os

Po přidání frekvenčních měničů do Sercos sítě a tím i zároveň do projektu, byly přidány, popřípadě nalezeny, servopohony, které jsou k těmto frekvenčním měničům připojeny. V prostředí IndraWorks jsou veškeré pohony označovány jako reálné osy. Pro nastavení parametrů os je potřeba se přepnout do parametrizačního módu P2.

Při použití absolutního pozicování s translační osou je potřeba určit referenční nulu enkodéru. Z důvodu absence jakéhokoliv referenčního snímače byla tato referenční nula nastavena ručně pro pozici kabiny výtahu v nejnižším podlaží, tedy v prvním patře. Pro nastavení zbývajících parametrů byly zvoleny jednotky v milimetrech, nebo popřípadě jednotky pro rychlosti v milimetrech za sekundu. Dalším parametrem, který je potřeba nastavit pro správné pozicování, je přepočítání jedné otáčky servopohonu na lineární posun – konstanta „k“. V tomto konkrétním případě bylo provedeno nastavení hodnoty udávané výrobcem podle typu použité řemenice, na které je umístěn pohyblivý řemen s kabinou výtahu.



Obrázek 15 Nastavení parametrů osy

5.3.1 Testování polohování

Před nastavením osy a samotným programováním úlohy bylo potřeba otestovat a vyzkoušet možnosti, které připadají v úvahu, a následně vybrat tu nejvhodnější. Na výběr máme ze čtyř možností – možnost rotační osy, a to buď s absolutním pozicováním, nebo pozicováním modulo, nebo lze zvolit použití osy translační s absolutním pozicováním, nebo pozicováním modulo. Každý typ polohování má své specifické vlastnosti a pro různé typy úloh se používá jiný způsob polohování. Z testovaných polohování byla vzhledem k povaze úlohy, která má vykonávat pohyb lineární, vybrána jako nejlepší možnost translační osa s absolutním pozicováním.

5.3.2 Nastavení limitů

Nastavení limitů slouží k omezení pohybu osy, nebo pro nastavení možné maximální rychlosti osy, anebo akcelerace. Koncové limity je potřeba správně nastavit, aby v případě problému nedošlo k zničení zařízení. Nastavení lze provést dvěma způsoby: podle automatického nastavení pomocí koncových snímačů připojených k frekvenčnímu měniči nebo k PLC automatu, nebo manuálně. Jelikož v úloze nebyly použity žádné pomocné koncové snímače, byly tyto limity nastavené pomocí manuálního pohybu os do krajních poloh a odečtením hodnot z enkodéru.

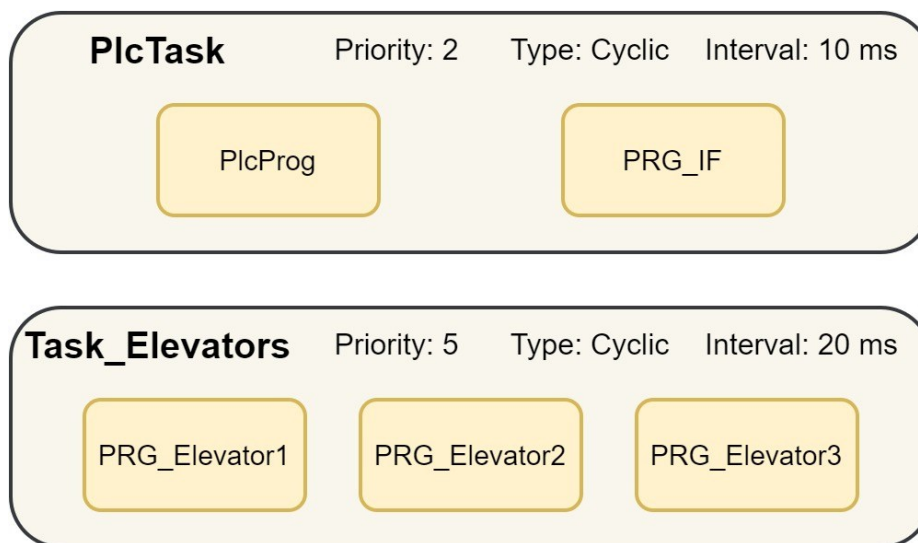
Motion limit values - Elevator1		
Elevator1		
	Setting the control	Drive settings
Position limit value monitoring	<input checked="" type="checkbox"/> Active	<input checked="" type="checkbox"/> Active
Actual position value	<input type="text" value="0"/>	<input type="text" value="0.0"/> mm
Position limit value, positive	<input type="text" value="1145"/>	<input type="text" value="1145.0"/> mm
Position limit value, negative	<input type="text" value="-5"/>	<input type="text" value="-5.0"/> mm
Travel range limit switch	<input type="checkbox"/> Active	<input type="checkbox"/> Active
Travel range limit switch operation mode:	<input type="radio"/> Opener <input checked="" type="radio"/> Closer	<input type="radio"/> Opener <input checked="" type="radio"/> Closer
Treating the travel range violation: - as error (according to the error reaction) - as fatal warning (standstill)	<input checked="" type="radio"/> Error <input type="radio"/> Warning	<input checked="" type="radio"/> Error <input type="radio"/> Warning
Velocity limit value, positive	<input type="text" value="150"/>	<input type="text" value="150.0"/> mm/s
Velocity limit value, negative	<input type="text" value="-150"/>	<input type="text" value="-150.0"/> mm/s
Velocity limit value, bipolar	<input type="text" value="150"/>	<input type="text" value="150.0"/> mm/s
Acceleration limit value, bipolar	<input type="text" value="1000"/>	<input type="text" value="1000.0"/> mm/s ²
Jerk limit value, bipolar	<input type="text" value="0"/>	<input type="text" value="0.0"/> mm/s ³
Torque/force value, positive	<input type="text" value="100"/>	<input type="text" value="100.0"/> %
Torque/force value, negative	<input type="text" value="-100"/>	<input type="text" value="-100.0"/> %
Torque/force limit value, bipolar	<input type="text" value="100"/>	<input type="text" value="100.0"/> %

Obrázek 16 Nastavení limitů osy

5.4 Části řídicí aplikace

Při tvorbě řídicí aplikace byl projekt rozdělen do několika samostatných programů. Projekt obsahuje program „PlcProg“, který byl využit pro inicializaci jednotlivých os, ale samotné řízení bylo rozděleno do dalších vytvořených programů. Pro řídicí aplikaci byly vytvořeny čtyři programy v projektu. Pro všechny vytvořené programy je použit jazyk „ST – Strukturovaný text“. Jeden z vytvořených programů slouží k vyhodnocování a řízení pohybu jednotlivých výtahů. Zbývající tři programy se starají o samotné řízení/pohyb jednotlivých výtahů. Logika vyhodnocení a řízení je podrobněji popsána v následujících kapitolách.

Pro volání jednotlivých programů, bylo využito cyklu programu pomocí taskového systému. Tento systém umožňuje vytvořit více tasků s různou prioritou a s jiným časem pro vykonání. Byly vytvořené dva tasky: jeden pro volání programu s inicializací os a program vyhodnocení a druhý vytvořený task slouží pro volání programu výtahů. Toto vzniklé rozdělení umožnilo vytvořit prioritní systém ve volání jednotlivých programů, kde byla nastavena vyšší priorita tasku s voláním programu vyhodnocení a inicializací, a tím byl vyřešen problém s vyhodnocováním požadavků od uživatele v průběhu vykonávání volaných funkcí výtahu. Největší problém nastával při průběhu použitého časovače „TON“, tedy časovače zpoždění, který je využíván pro určení času do zavření dveří výtahu. V tomto čase docházelo ke špatnému vyhodnocení ostatních požadavků pro jednotlivé výtahy. Tento problém byl odstraněn rozdělením programů do taskového systému. Toto rozdělení a nastavení taskového systému je graficky znázorněno na obrázku 17.



Obrázek 17 Rozdělení taskového systému

5.4.1 Definice proměnných

V projektu byly použity lokální i globální proměnné, hlavní rozdíl mezi proměnnými je jejich viditelnost v rámci projektu. Globální proměnné jsou viditelné v rámci celého projektu v PLC a lokální proměnné jsou viditelné pouze v části programu, kde jsou definovány.

Globální proměnné

Pro práci s virtuálními tlačítky v uživatelském rozhraní, byla v programu vytvořena struktura „ST_Buttons“. Jedná se o strukturu čtyř „bool“ polí o osmi prvcích pro každé tlačítko výtahu nebo podlaží. Tato struktura byla nadefinována jako globální proměnná, aby byla k ní možnost přístupu ze všech programů v projektu. Další důležitým vytvořeným polem, které bylo také definováno jako globální proměnná, bylo pole osmi prvků, tentokrát celočíselného typu. V tomto poli byly nadefinovány absolutní pozice os pro jednotlivé podlaží výtahu a poté podle příslušného zvoleného podlaží je z pole hodnota vytáhnutá a přiřazena servopohonu, jako posun na pozici. Další použité globální proměnné jsou proměnné, které nás informují o pozici jednotlivých výtahu, o stavu výtahu, popřípadě dveří, jejich seznam a specifikace je uvedena v tabulce 5 níže.

Tabulka 5 Seznam použitých globálních proměnných v projektu

Název proměnné	Datový typ	Popis
Buttons	ST_Buttons	Struktura tlačítek
Position_Elevator	ARRAY	Pole pozic výtahu
Move_Elevator1	BOOL	Informace o pohybu výtahu 1
Request_Elevator1	UINT	Požadavek posunu výtahu 1
Buttons_Empty_El1	BOOL	Informace o stisknutých tlačítkách v kabině výtahu 1
Memory_Elevator1	UINT	Paměť pro vykonání pohybu požadavků z podlaží – výtah 1
Move_Elevator2	BOOL	Informace o pohybu výtahu 2
Request_Elevator2	UINT	Požadavek posunu výtahu 2
Buttons_Empty_El2	BOOL	Informace o stisknutých tlačítkách v kabině výtahu 2
Memory_Elevator2	UINT	Paměť pro vykonání pohybu požadavků z podlaží – výtah 2
Move_Elevator3	BOOL	Informace o pohybu výtahu 3
Request_Elevator3	UINT	Požadavek posunu výtahu 3
Buttons_Empty_El3	BOOL	Informace o stisknutých tlačítkách v kabině výtahu 3
Memory_Elevator3	UINT	Paměť pro vykonání pohybu požadavků z podlaží – výtah 3

Lokální proměnné

Lokální proměnné jsou použité v jednotlivých programech projektu. Za zmínku stojí využitý časovač, který je definován ve všech třech programech výtahu. Jedná se o časovač „TON“, tedy o zpoždění. Toto zpoždění je použito pro čas, po kterém má dojít k zavření dveří výtahu. Zbýlé použité lokální proměnné jsou uvedeny v tabulce 6 níže.

Tabulka 6 Seznam použitých lokálních proměnných v projektu

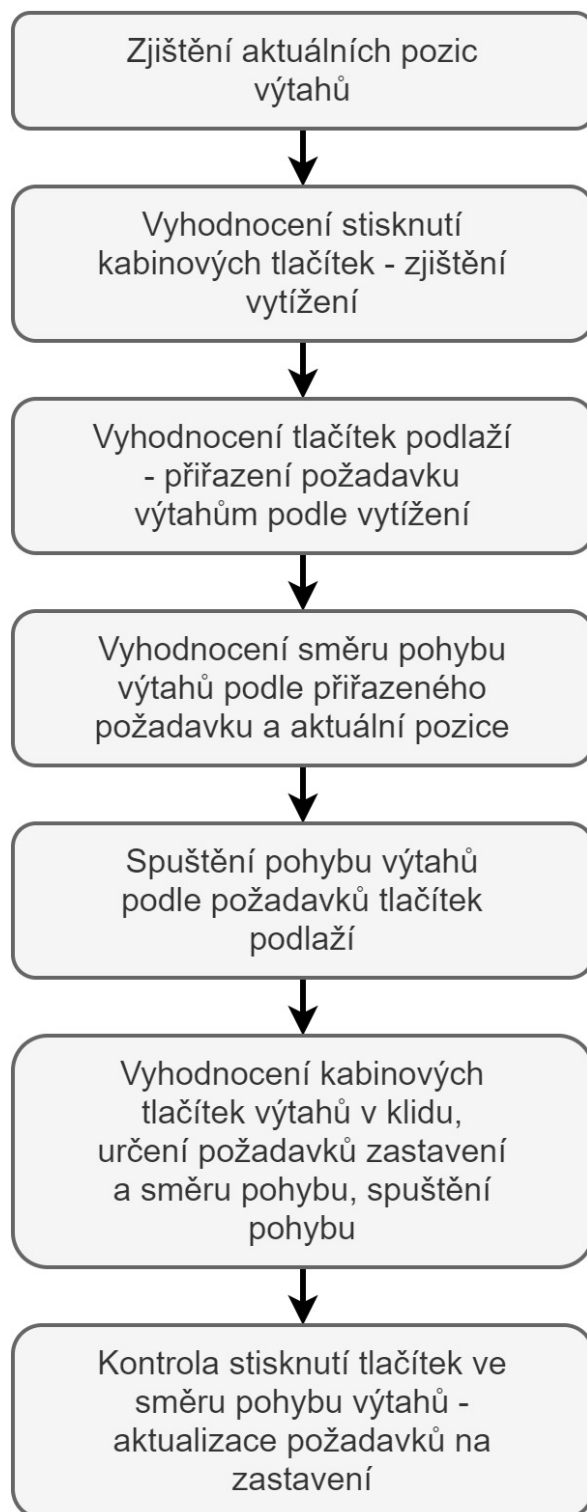
Název proměnné	Datový typ	Popis
Program „PRG_IF“		
i	UINT	Proměnná pro cyklus „FOR“
Position_Elevator1	UINT	Informace o aktuální pozici výtahu 1
Index_Elevator1	UINT	Index stisknutého tlačítka v kabině výtahu 1
Direction_Elevator1	BOOL	Určení směru pohybu výtahu 1
Position_Elevator2	UINT	Informace o aktuální pozici výtahu 2
Index_Elevator2	UINT	Index stisknutého tlačítka v kabině výtahu 2
Direction_Elevator2	BOOL	Určení směru pohybu výtahu 2
Position_Elevator3	UINT	Informace o aktuální pozici výtahu 3
Index_Elevator3	UINT	Index stisknutého tlačítka v kabině výtahu 3
Direction_Elevator3	BOOL	Určení směru pohybu výtahu 3
Program „PRG_Elevator1“		
iStep_Elevator1	UINT	Proměnná stavového automatu pro výtah 1
Door_Elevator1	BOOL	Informace o stavu dveří výtahu 1
Door_Timer1	TON	Definice časovače zpoždění pro zavření dveří výtahu 1
Program „PRG_Elevator2“		
iStep_Elevator2	UINT	Proměnná stavového automatu pro výtah 2
Door_Elevator2	BOOL	Informace o stavu dveří výtahu 2
Door_Timer2	TON	Definice časovače zpoždění pro zavření dveří výtahu 2
Program „PRG_Elevator3“		
iStep_Elevator3	UINT	Proměnná stavového automatu pro výtah 3
Door_Elevator3	BOOL	Informace o stavu dveří výtahu 3
Door_Timer3	TON	Definice časovače zpoždění pro zavření dveří výtahu 3

5.4.2 Program pro vyhodnocování

Program pro vyhodnocení v projektu figuruje pod názvem „PRG_IF“. V programu dochází k veškerému rozhodnutí o pohybu jednotlivých os. V tomto programu dochází k rozhodnutím pro všechny tři výtahy, ale výkonné části programu jsou totožné, akorát s použitím jiných proměnných určených právě pro daný výtah. V následujícím textu bude popsán vyhodnocující proces pro jeden z výtahu, konkrétně pro výtah číslo jedna.

Na začátku programu jsou pomocí cyklu „FOR“, který je indexován od 1 do 8, zjišťovány informace. První je zjištěno pomocí porovnání aktuální pozice výtahu z funkce „AxisData“ s polem „Position_Elevator“ aktuální pozice výtahu a uložena do proměnné „Position_Elevator1“. Další částí cyklu je vyhodnocení stisknutých kabinových tlačítek a zjištění tak vytížení výtahu (proměnná: „Buttons_Empty_El1“). V cyklu dochází také k vyhodnocení podlažních tlačítek a v případě zjištění stisku tlačítka dojde k rozhodnutí podle vytížení výtahu a přiřazení požadavku pro daný výtah (proměnná: „Memory_Elevator1“). Poslední fází cyklu je porovnání aktuální pozice výtahu (proměnná: „Position_Elevator1“) s požadavkem pohybu (proměnná: „Memory_Elevator1“), kdy dojde k vyhodnocení směru pohybu výtahu (proměnná: „Direction_Elevator1“).

Po dokončení zjišťování všech důležitých informací probíhá další část programu, která už zajišťuje pokyny pro programy starající se o řízení servopohonu, tedy „PRG_Elevator“ a příslušné číslo výtahu. Zde je podle vyhodnocených informací o stisku spuštěn pokyn pro pohyb výtahu, nastavení proměnné „Move_Elevator1“ a přiřazení požadavku o stisku podlažního tlačítka do proměnné „Request_Elevator1“ pro vykonání pohybu. V případě absence požadavků na pohyb ze strany podlažních tlačítek jsou zde vyhodnoceny požadavky z kabinových tlačítek při výtahu v klidu, a tím spojené určení směru pohybu výtahu a samotné spuštění pohybu na požadavek kabinového tlačítka. V případě pohyby výtahu jsou zde vyhodnocovány a hledány stisknutá tlačítka ve směru pohybu výtahu a při nalezení je aktualizován požadavek na zastavení (proměnná „Request_Elevator1“). Všechny tyto popsané fáze vyhodnocení se zde nachází pro všechny tři výtahy. Grafické znázornění jednotlivých popsaných fází programu vyhodnocení je znázorněno na obrázku 18.

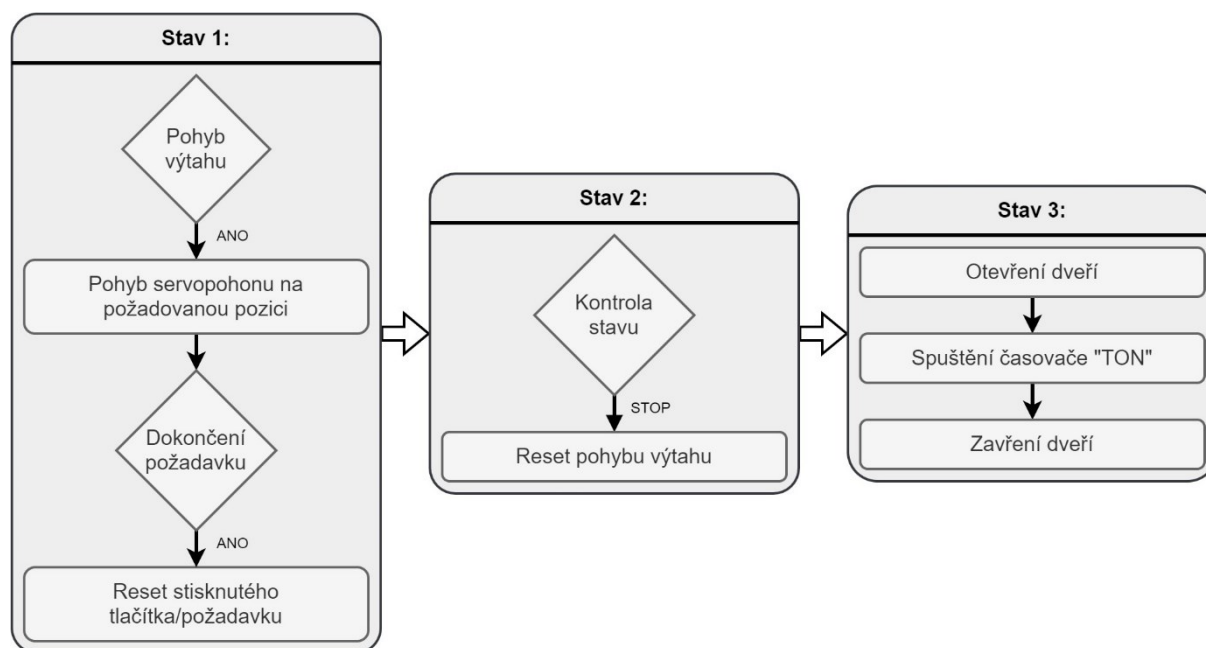


Obrázek 18 Blokové schéma programu pro vyhodnocení

5.4.3 Program pro řízení servopohonu/výtahu

Program pro řízení servopohonu existuje v projektu pro každý výtah zvlášť, tedy třikrát, ale jejich logika je totožná. Jedná se o programy s názvem „PRG_Elevator“ a příslušné číslo výtahu. V následujícím textu bude popsán program pro výtah číslo jedna.

Pro program je vytvořen stavový automat pomocí cyklu „switch“, který se skládá celkem ze tří stavů. V prvním stavu tohoto programu je vyhodnocena podmínka spuštění pohybu daného výtahu (proměnná: „Move_Elevator1“) a v případě kladného vyhodnocení dojde k přiřazení požadované polohy pomocí vyčtení hodnoty z pole (pole: „Position_Elevator[Request_Elevator1]“) do funkce pro absolutní posun servopohonu „arAxisCtrl_gb“ a dojde k spuštění pohybu osy. Tato požadovaná poloha se může v průběhu pohybu osy změnit, podle daného vyhodnocení z programu „PRG_IF“. Po dokončení požadavku vyhodnocení pomocí podmínky porovnání aktuální pozice výtahu z funkce „AxisData“ s požadovanou polohou je vyresetováno stisknuté tlačítko/požadavek ze struktury tlačítek (proměnná: „Buttons.Buttons_Elevator1[Request_Elevator1]“, a v případě, že byl požadavek pohybu pro tlačítko podlažní, tak proměnná: „Buttons.Buttons_Floor[Request_Elevator1]“) a program přejde do stavu číslo dva. V druhém stavu je kontrolován stav servopohonu pomocí funkce „arAxisStatus“, z které jsme schopni zjistit, zda se osa nachází v klidu a nedochází již k žádnému pohybu. V případě kladného vyhodnocení dojde k vyresetování proměnné pohyb výtahu (proměnná: „Move_Elevator1“) a dále program přejde do posledního stavu, tedy do stavu číslo tři. Pokud by podmínka nebyla vyhodnocena kladně, dojde k nedokončení pohybu, a tedy do nevyžádaného stavu, který by bylo nutné odstranit zásahem obsluhy. Ve stavu tři dojde k otevření dveří, tedy nastavení proměnné „Door_Elevator1“ a spuštění časovače zpoždění „TON“ (proměnná „Door_Timer1“). Po uplynutí nastaveného času pěti sekund dojde k zavření dveří výtahu, tedy reset proměnné „Door_Elevator1“. Dojde také k přechodu do prvního stavu programu, kdy opět následuje čekání na pokyn k pohybu výtahu. Na následujícím obrázku 19, jsou graficky zpracovány jednotlivé stavy tohoto programu.



6 Návrh a realizace uživatelského rozhraní

Pro návrh realizace uživatelského rozhraní byl použit, stejně jako pro tvorbu řídicí aplikace, software BOSCH Rexroth Indra Works Engineering. Na výběr byl ještě software specializovaný přímo na tvorbu vizualizace – externí program WinStudio, který obsahuje více možností a funkcí. Vzhledem k ne tak náročnému uživatelského rozhraní pro použité komponenty byl vyhodnocen jako dostačující program Indra Works Engineering s implementovanou knihovnou objektů WinStudio Screen Tools.

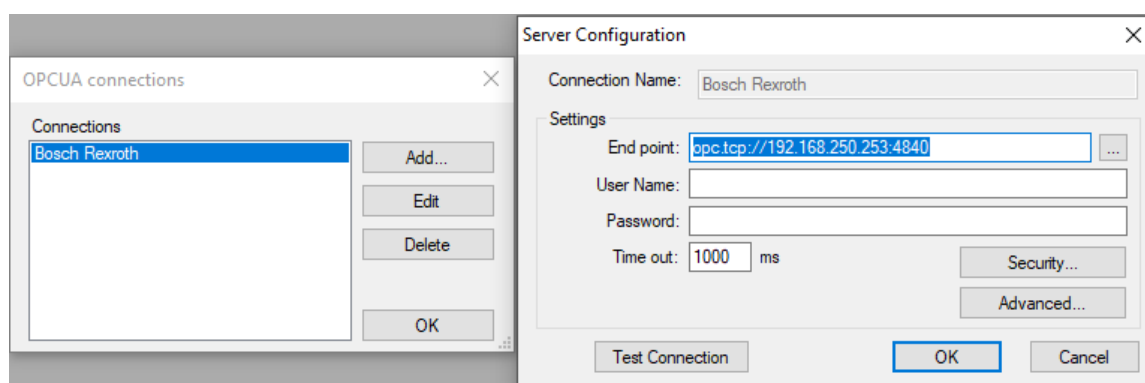
Uživatelské rozhraní bylo vytvořeno pro ovládací panel výtahu suplováním nepoužitých fyzických tlačítek výtahu v podobě digitálních vstupů do PLC. Toto řešení bylo navrženo vzhledem k malému počtu digitálních vstupů na použitém PLC.

6.1 HW konfigurace HMI panelu

Pro vytvoření uživatelského rozhraní bylo potřeba do konfigurace přidat HMI panel, který bude zobrazovat vizualizaci. K vytvořené úloze není připojen fyzicky HMI panel, a proto bude vizualizace spuštěna na připojeném notebooku. Další možností, která se nabízela, je použít HMI panel z výukových stolů v budově CPIT TL3. Vzhledem ke spuštění vizualizace na notebooku se nemusí přidávat přesný použitý model HMI panelu. Přidán byl tedy panel z knihovny s označením VH21/VR21, konkrétně model VH21. IP adresa se zde nemusí nastavovat a zůstane jako localhost. Jedná se o devíti palcový panel s rozlišením 1280x800 pixelů. Devíti palcový HMI panel byl vybrán z důvodu možného použití navrhované vizualizace právě na panelech u výukových stolů v budově. Tímto je hardwarová konfigurace panelu hotová a je třeba vytvořit komunikaci pro spojení panelu s PLC, která je popsána v následující kapitole.

6.2 Konfigurace komunikace

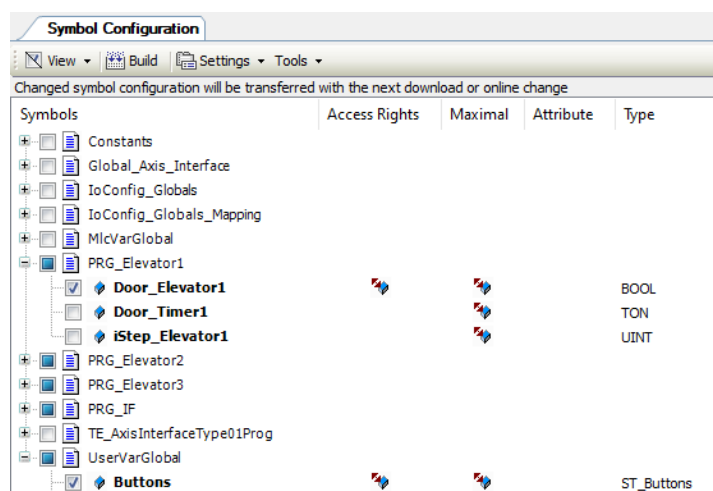
Komunikace mezi PLC a HMI panel slouží pro přístup k použitým proměnným z programu v PLC ve vizualizaci. Zde se nabízí více možností komunikací, např. pomocí OPC, DDE, OPC-UA a TCP/IP. Nakonec byla využita komunikace skrze OPC-UA server v ně PLC. Byla tedy přidána nová komunikace a s tím spojená rekonfigurace serveru. Pro server byla nastavena adresa `opc.tcp://192.168.250.253:4840`, jedná se o IP adresu, která byla přiřazena PLC.



Obrázek 20 Konfigurace OPC-UA serveru

6.3 Použité tagy ve vizualizaci

Zpřístupnění jednotlivých proměnných je potřeba povolit pomocí Symbol Configuration v projektovém stromě, zde je zapotřebí zaškrtnout všechny proměnné, které budou použity v HMI, viz obrázek níže.



Obrázek 21 Symbol configuration

Pro vytvoření uživatelského ovládacího panelu bylo zapotřebí přenést pomocí OPC-UA komunikace následující tagy:

Tabulka 7 Seznam použitých tagu ve vizualizaci

Název tagu	Datový typ	Popis
Buttons	ST_Buttons	Struktura tlačítek
Door_Elevator1	BOOL	Informace o stavu dveří výtahu 1
Door_Elevator2	BOOL	Informace o stavu dveří výtahu 2
Door_Elevator3	BOOL	Informace o stavu dveří výtahu 3
Move_Elevator1	BOOL	Informace o pohybu výtahu 1
Move_Elevator2	BOOL	Informace o pohybu výtahu 2
Move_Elevator3	BOOL	Informace o pohybu výtahu 3
Position_Elevator1	UINT	Informace o aktuální pozici výtahu 1
Position_Elevator2	UINT	Informace o aktuální pozici výtahu 2
Position_Elevator3	UINT	Informace o aktuální pozici výtahu 3

6.4 Použité objekty a funkce

Pro vytvoření uživatelského rozhraní byly použity objekty z knihovny WinStudio Screen Tools, která je importována přímo v programu IndraWorks Engineering. V případě nutnosti použití jiných objektů je potřeba použít pro tvorbu vizualizace pro HMI panel jiné knihovny s prvky, které obsahují program WinStudio.

Použité objekty:

Text	Objekt s pevným textem s možností zobrazení dat z tagu v textovém formátu, používán pro specifikaci jednotlivých tlačítek a určení pozice výtahu.
Rectangle	Vizualizační obdélníkový objekt, využit jako vizuální rozdělení jednotlivých částí.
Rounded rectangle	Vizualizační zaoblený obdélníkový objekt, použit pro znázornění podlažních tlačítek a indikaci dveří výtahu.
Ellipse	Vizualizační objekt elipsy, použit pro znázornění výtahových tlačítek.

S použitím jednotlivých nabízených funkcí je možnost použitým objektům dodat funkce, jako například reakci na stisknutí objektu, změnu barvy v případě změny hodnoty příslušného tagu a další.

Použité funkce:

Activate/deactivate command	Funkce pro nastavení chování objektu v případě stisku je možné změnit hodnotu přiřazeného tagu, otevřít/zavřít novou obrazovku vizualizace, popřípadě pomocí VBScriptu napsat logiku chování. V tvorbě uživatelského rozhraní byla využita funkce „SetTag“ pro jednotlivá tlačítka výtahu.
Activate/deactivate color change	Jedná se o funkci reagující na změnu příslušného přiřazeného tagu a pomocí hodnot tagu reaguje objekt s touto funkcí změnou barvy podle nastavení, popřípadě rozblikáním objektu v příslušné barvě. Toto chování bylo využito pro znázornění stisku jednotlivých tlačítek, informací o stavu dveří výtahu a o stavu pohybu výtahu pomocí blikání objektu.

6.5 Výsledná realizace uživatelského rozhraní

Navržená výsledná realizace uživatelského rozhraní pro ovládání výtahů je vizuálně rozdělena do čtyř bloků. První tři bloky při pohledu zleva slouží pro ovládání jednotlivých výtahů z kabiny a poslední blok umístěn vpravo je navržen pro ovládání výtahu z příslušných podlaží. V horní části bloku pro ovládání jednotlivých výtahů, se nachází název výtahu, vedle kterého je pak aktuální pozice výtahu v podobě číslovky, za kterou je zaoblený obdélník, který v případě pohybu výtahu začne blikat zelenou barvou (na přiloženém obrázku 22 je statické podbarvení zelenou barvou). Největší část tohoto bloku zabírají jednotlivé objekty s čísly příslušných podlaží sloužících jako tlačítka pro ovládání výtahu z kabiny. V případě požadavku na zastavení v příslušném patře je možnost tato tlačítka stisknout a indikace požadavku již stisknutého tlačítka je znázorněna podbarvením objektu modrou barvou. V případě vykonání tohoto požadavku je barva objektu navracena do původního šedého odstínu. Poslední částí bloku, a to ve spodní části, je indikace otevření nebo zavření dveří výtahu. Tato indikace je znázorněná pomocí zaobleného obdélníku, který v případě zavřených dveří má barvu červenou a v případě otevřených dveří změni svou barvu na zelenou.

Blok sloužící pro ovládání výtahů z jednotlivých podlaží obsahuje pouze osm tlačítek, které jsou reprezentovány objektem zaobleného obdélníku obsahující text, který definuje, o které podlaží se jedná. Tyto objekty slouží jako tlačítka k přivolání výtahu. V případě požadavku o přivolání výtahu do příslušného patra je možné tlačítko stisknout a indikace požadavku již stisknutého tlačítka je znázorněna podbarvením objektu do modré barvy.



Obrázek 22 Ovládací panel výtahů

Závěr

Bakalářská práce se věnovala rozboru možností polohování servopohonů a jejich řízení pomocí programovatelného automatu. V úvodní části jsme se seznámili se samotným složením servopohonu, a to hlavně části enkodéru, který se stará o dodávání informací ohledně polohy řízené osy. Dále jsme se zabývali doplňujícími senzory, které mohou pomáhat s určením polohy a samotnou možností řízení pomocí frekvenčních měničů. Taktéž se práce věnovala organizaci PLCopen, jenž vytváří standardy pro tvorbu řídicích aplikací a rovněž vytváří předpřipravené knihovní funkce pro pohodlnější tvorbu programů. Pozornost je směřována na detailnější popis pohybových úloh.

Pro otestování možností polohování servopohonu byla navržena a vyrobena laboratorní instalace, která je umístěna a zakomponována v místnosti Smartfactory – CPIT TL3. V navržené instalaci byly použity komponenty od firmy Bosch a byla navržena a konstruována tak, aby bylo možné výslednou instalaci používat i během výuky. Pro navrženou úlohu jsme také vyzkoušeli zhodnocení rizika pro zajištění funkční bezpečnosti za použití definované normy ČSN ISO 13849-1, která obsahuje stanovení Performance úrovně. Výsledná bezpečnostní úroveň vytvořené laboratorní instalace je PL_a – jedná se o nejnižší neboli taky nejbezpečnější úroveň.

Po vytvoření laboratorní instalace byly za použití vývojového prostředí BOSCH Rexroth Indra Works Engineering otestovány všechny dostupné možnosti polohování. Jako výsledná řídicí aplikace pro ukázkou možnosti polohování byla zvolena soustava tří simulovaných výtahu vytvořená pomocí použitých řemenů na osách servopohonu. Pro výslednou řídicí aplikaci bylo z testovaných možností vybráno polohování pomocí translační osy s absolutním pozicováním. Tento typ polohování byl vybrán převážně z důvodu vykonávání lineárního pohybu osy, což zvolený typ umožňuje. Pro použití tohoto polohování je důležité stanovit referenční nulu enkodéru. Z důvodu absence jakéhokoliv referenčního snímače bylo nutné provést nastavení ručně.

Pro vytvořenou řídicí aplikaci bylo také vytvořeno uživatelské rozhraní pomocí vizualizace na HMI panelu. Z důvodu absence HMI panelu na vytvořené instalaci byl pro spuštění vizualizace využit připojený notebook. Vytvořené uživatelské rozhraní obsahuje prvky pro ovládání simulovaných výtahu a taktéž informuje uživatele o stavech, ve kterých se daný výtah nachází. Výsledkem práce je plně funkční řídicí aplikace tří simulovaných výtahů s přívětivým ovládacím panelem celé úlohy.

V rámci zdokonalení navržené laboratorní instalace je zapotřebí doplnění úlohy o referenční snímače pro větší pohodlí při práci s osami, zejména pro zpřesnění určení jejich pozice. Dále by bylo vhodné laboratorní instalaci připojit do sítě použité v místnosti Smartfactory v budově CPIT TL3, a tím vyřešit problém s absencí HMI panelu na laboratorní instalaci. V místnosti by poté bylo možné spouštět vytvořenou vizualizaci na jakémkoliv HMI panelu, například na panelech umístěných na pracovních stolech v sousední části.

Literatura

- [1] *Servo motor* [online]. [cit. 2020-10-15]. Dostupné z: <https://www.andmotor.com/servo-motor/>
- [2] *Rotační enkodéry pro řízení pohonů a pohybu* [online]. [cit. 2020-10-20]. Dostupné z: <https://automatizace.hw.cz/rotacni-ekodery-pro-rizeni-pohonu-a-pohybu.html>
- [3] *Rotační enkodéry ELTRA* [online]. [cit. 2020-10-20]. Dostupné z: <http://www.odbornecasopisy.cz/elektro/casopis/tema/rotacni-ekodery-eltra--13671>
- [4] *Inkrementální enkodér* [online]. [cit. 2020-10-20]. Dostupné z: <https://robotika.vosrk.cz/guide/sensors/decode/cs>
- [5] *FREKVENČNÍ MĚNIČE A VŠE O NICH* [online]. [cit. 2020-11-02]. Dostupné z: <https://www.elektromotory.cz/frekvencni-menice-proc-a-jak>
- [6] *Servopohony* [online]. [cit. 2020-11-02]. Dostupné z: <https://www.invtcz.cz/servopohony>
- [7] *Snímače polohy* [online]. [cit. 2020-11-02]. Dostupné z: <http://sps.hellebrand.cz/text0910/mch/senzorika-poloha.pdf>
- [8] *Indukční čidlo, M12, 0-2 mm, PNP/NO, kabel LJ12A3-2-Z/BY* [online]. [cit. 2020-11-25]. Dostupné z: <https://www.gme.cz/indukcni-cidlo-lj12a3-2-z-by-m12-0-2-mm-pnp-no-kabel>
- [9] *Lineární snímače polohy* [online]. [cit. 2020-11-02]. Dostupné z: <https://www.megatron.cz/linearni-snimace-polohy>
- [10] *What is PLCopen* [online]. [cit. 2020-11-13]. Dostupné z: <https://plcopen.org/what-plcopen>
- [11] *Benefits of membership* [online]. [cit. 2020-11-13]. Dostupné z: <https://plcopen.org/benefits-membership>
- [12] *IEC 61131-3: a standard programming resource* [online]. [cit. 2020-11-20]. Dostupné z: <http://www.oacg.co.uk/plcopen.pdf>
- [13] *Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys)* [online]. [cit. 2020-11-20]. Dostupné z: <https://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-3-codesys>
- [14] *Ladder diagramy – základy* [online]. [cit. 2020-11-20]. Dostupné z: <http://fatek.seapraha.cz/2008/03/24/zakladr/>
- [15] *PLC Programming according to IEC 61131-3* [online]. [cit. 2020-11-20]. Dostupné z: <https://www.eclipse.org/4diac/documentation/html/before4DIAC/iec61499.html>
- [16] *Technical activities* [online]. [cit. 2020-11-13]. Dostupné z: <https://plcopen.org/technical-activities>
- [17] *Logic* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/logic>
- [18] *Safety* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/safety>
- [19] *Communication* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/communication>
- [20] *XML Exchange* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/xml-exchange>
- [21] *PLCopen XML now available as IEC 61131-10* [online]. [cit. 2020-11-17]. Dostupné z: https://plcopen.org/sites/default/files/downloads/plcopen_xml_exchange.pdf

- [22] *Certification* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/certification>
- [23] *Motion Control* [online]. [cit. 2020-11-17]. Dostupné z: <https://plcopen.org/technical-activities/motion-control>
- [24] *PLCopen Motion Control* [online]. [cit. 2020-11-17]. Dostupné z: https://plcopen.org/sites/default/files/downloads/plcopen_motion_control_overview.pdf
- [25] *PLCopen State Diagram* [online]. [cit. 2020-11-20]. Dostupné z: https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/MotCoLib/MotCoLib/General_Description_of_Motion_Control_Libraries/General_Description_of_Motion_Control_Libraries-4.htm
- [26] *IndraControl L45* [online]. [cit. 2020-12-15]. Dostupné z: https://www.boschrexroth.com/dcc/content/internet/en/pdf/PDF_p150096_en.pdf
- [27] *MS2N Synchronous servo motors* [online]. [cit. 2021-01-10]. Dostupné z: <https://www.cmafh.com/images/Master%20PDFs/BRC/Motors/MS2N/Rexroth%20MS2N%20Product%20Catalog.pdf>
- [28] *Rexroth IndraDrive Cs Drive Systems with HCS01* [online]. [cit. 2021-01-10]. Dostupné z: <https://www13.boschrexroth-us.com/goto/edc/GoTo-IndraDrive-Cs.pdf>
- [29] *Bezpečnost strojů - 2. díl – PL vs. SIL* [online]. [cit. 2021-04-15]. Dostupné z: <https://automatizace.hw.cz/bezpecnost-stroju/bezpecnost-stroju-2-dil-pl-vs-sil.html>

Seznam příloh

Součástí práce jsou přílohy v elektronické podobě.

- Projekt v programu BOSCH Rexroth Indra Works Engineering
 - BP_VAV0174.xiwp
- Dokumenty
 - Konstrukce_boxu_Alutec.pdf
 - Zapojeni_ulohy_Eplan.pdf
 - DataSheet_PLC_L45-L65-L85.pdf
 - DataSheet_servopohon_MS2N.pdf
 - DataSheet_frekvencni_menic_HCS01.pdf